BBOB-Benchmarking the Generalized Generation Gap Model with Parent Centric Crossover

Petr Pošík
Czech Technical University in Prague, Faculty of Electrical Engineering, Dept. of Cybernetics
Technická 2, 166 27 Prague 6
posik@labe.felk.cvut.cz

ABSTRACT

The generalized generation gap (G3) model of an evolutionary algorithm equipped with the parent centric crossover (PCX) is tested on the BBOB 2009 benchmark testbed. To improve its behavior on multimodal functions a multistart strategy is used. The algorithm shows promising results especially for the group of 'moderate' functions, but fails systematically on the group of 'multimodal' functions.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global Optimization, Unconstrained Optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms, Experimentation, Performance, Reliability

Keywords

Benchmarking, Black-box optimization, Evolutionary computation, Generalized generation gap, Parent centric crossover

1. INTRODUCTION

The generalized generation gap (G3) model of evolutionary algorithm was introduced by Deb in [1]. In the same article, the G3 model was used with the parent centric crossover (PCX) operator introduced in [2]. In the mentioned article, the combination of G3 and PCX was tested on three 20-dimensional functions and was shown to be more efficient than the state-of-the-art evolutionary strategy with covariance matrix adaptation (CMA-ES) [6]. G3 model with the PCX operator was also an order of magnitude better than G3 with other crossover operators (simulated binary crossover, unimodal normally distributed crossover, or simplex crossover).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-505-5/09/07 ...$5.00.

The purpose of this article is to provide a better comparison of G3+PCX on a wider set of functions and wider set of search-space dimensionalities.

2. ALGORITHM DESCRIPTION

2.1 Generalized Generation Gap (G3)

G3 is an elitist steady-state model which operates as follows (based on [1]):

1. From the population $\mathcal{B}$ of size N, select the set of parents $\mathcal{P}$, i.e. select the best member and $\mu - 1$ other individuals uniformly.
2. Generate the set $\mathcal{C}$ of λ offspring from the μ selected parents $\mathcal{P}$ using a chosen recombination scheme (PCX in this article).
3. Choose a subset $\mathcal{R}$ of r candidates for replacement uniformly from the population $\mathcal{B}$.
4. Replace the subset $\mathcal{R}$ of population $\mathcal{B}$ with the best r members from the combined set $\mathcal{C} \cup \mathcal{R}$.
5. Iterate all steps until termination condition is satisfied.

2.2 Parent Centric Crossover (PCX)

The PCX operator was first introduced in [2] as an extension of the simulated binary crossover for any number of parents. In PCX, one parent, say $\mathbf{x}_p$, is chosen as the main one which will be used as the center for the distribution for sampling one offspring (for each offspring the main parent is chosen with equal probability). Let the other $\mu - 1$ parents be $\mathbf{x}_1, \ldots, \mathbf{x}_{\mu-1}$. First, the mean vector $\mathbf{g}$ of all μ parents is computed. Then, the direction vector $\mathbf{d}_p = \mathbf{x}_p - \mathbf{g}$ is computed. This vector has a significant role in the PCX, since it represents the main direction which is dealt with in a different way than with the other directions. For all other $\mu - 1$ parents, their perpendicular distances D_i, $i \in \langle 1, \mu - 1 \rangle$, to the $\mathbf{d}_p$ vector are computed. Based on them, the average distance $\bar{D}$ is found. The offspring is then created as

$$\mathbf{y} = \mathbf{x}_p + w_p \mathbf{d}_p + \sum_{i=1}^{l-1} w_i \bar{D} \mathbf{e}_i, \tag{1}$$

where $w_p \sim N(0, \sigma_p)$ is normally distributed random value with standard deviation σ_p and $w_i \sim N(0, \sigma_o)$ are normally distributed random values with standard deviation σ_o (common to all w_i). The vectors $\mathbf{e}_1, \ldots, \mathbf{e}_{l-1}$ are unit orthonormal base vectors spanning the linear subspace perpendicular to the vector $\mathbf{d}_p$.

In other words, the offspring is created from a l-dimensional normal distribution centered around the main parent $\mathbf{x}_p$ which is isotropic in all directions except the direction $\mathbf{d}_p$. The variance in all directions perpendicular to $\mathbf{d}_p$ is given by the standard deviation σ_o and by the average distance of the other parents to the $\mathbf{d}_p$ vector, while variance in the direction of the $\mathbf{d}_p$ vector is given by the distance of the main parent $\mathbf{x}_p$ from the parent mean $\mathbf{g}$ and by the standard deviation σ_p. A demonstration of generating 10,000 points using PCX with 3 parents can be seen in Fig. 1.

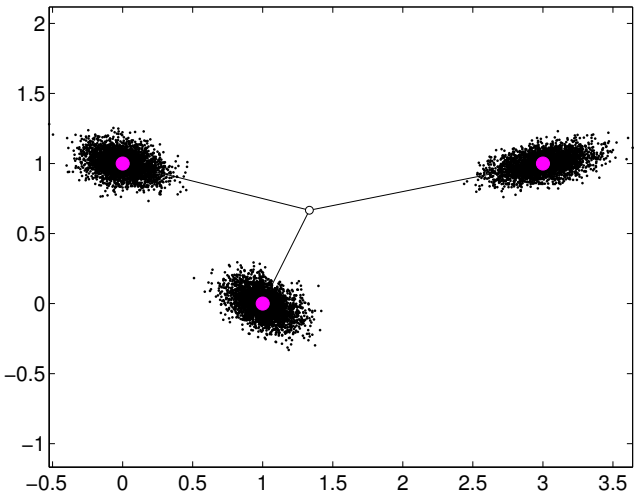

Figure 1: Demonstration of the offspring distribution induced by the PCX operator (10,000 points).

2.3 Parameter Settings

The G3+PCX algorithm has a lot of parameters, for the most of them the values suggested in [1] are used, i.e. for the G3 model

- number of parents $\mu = 3$,
- number of offspring $\lambda = 2$,
- number of replacement candidates $r = 2$,

and for the PCX operator, both standard deviations $\sigma_p = \sigma_o = 0.1$.

An important fact is that a highly local-search intensive variant of PCX operator was used: the main parent $\mathbf{x}_p$ is not selected randomly, but rather the best parent (and thus the best population member) is always chosen as $\mathbf{x}_p$.

A special care should be given to the population size setting. In [1], a population size of 100 was used for all three 20-dimensional functions. It was also shown that the performance of the G3+PCX drops quickly if insufficient population size is used, while it drops only slowly if larger than optimal population size is used. Several short experiments revealed that even for 2-dimensional functions, pretty large population size (about 90) is needed to find the solution reliably, while to solve the 40-dimensional Sphere function, a population size about 300 is needed. Thus, the following equation was used as the population sizing model:

$$N = 90 + 5 \cdot l. \tag{2}$$

This is rather conservative choice and the model is by no means optimal (for 20-dimensional space it gives $N = 190$ which is almost twice as much as the optimal value suggested in [1]). It is thus well possible that even better results can be provided by G3+PCX algorithm.

To improve performance on multimodal functions, a restarting strategy (see Fig. 2) was used.

2.4 Box Constraints? Yes

In the BBOB comparison, several global search algorithms are present. They usually require to set the box constraints to limit the search space. To be fair to algorithms used for unconstrained optimization, it was tested if the box constraints cannot be of any help also for them.

In [4], it was stated that the global optimum of many benchmark functions lies in $\langle -4, 4\rangle^D$ (where D is the search space dimensionality), and that all the benchmark functions should have their optimum in $\langle -5, 5\rangle^D$. This hypercube can thus serve as the bounding box for the optimization, i.e. when the algorithm wants to sample a point outside this box, it is forced to generate a new candidate.

1. The G3+PCX algorithm was run with box constraints $\langle -5, 5\rangle^D$, but this setting resulted in a poor performance on the simple Linear slope function—it is very hard for the algorithm to approach the solution lying on the boundary only from one side, while it might be simpler if it was allowed to overshoot the boundary and then focus on it.

2. The algorithm was then run completely unconstrained and the results on the Linear slope function were much better. But, the results for many other functions were worse than in the bounded case since the algorithm spent a lot of available budget on evaluating the unnecessary distant points.

3. The final configuration presented in this paper is to use the $\langle -6, 6\rangle^D$ hypercube as the search space. It is a viable compromise between the two previous cases.

2.5 The Crafting Effort

All parameter settings presented above are the same for all benchmark problems; the crafting effort CrE= 0.

2.6 Invariance Properties

G3+PCX algorithm is *invariant with respect to translation and rotation* as can be seen in the next section on the graphs for unrotated–rotated versions of Ellipsoid (2 and 10), Rastrigin (3 and 15), or Rosenbrock (8 and 9) functions. The algorithm uses fitness values only in the rank-based selection, thus it is also *invariant with respect to order-preserving transformations of the fitness function.*

3. EXPERIMENTAL PROCEDURE

The standard experimental procedure of BBOB was adopted: the algorithm was run on 24 test functions, 5 instances each, 3 runs on each instance. Each run was finished

- after finding a solution with fitness difference $\Delta f \leq 10^{-8}$, or
- after performing more than $5 \cdot 10^4 \cdot l$ function evaluations.

Each individual launch of the basic G3+PCX algorithm was interrupted (and the algorithm was restarted)

```
function [x, ilaunch] = bbobRestG3PCX(FUN, ...
    DIM, ftarget, outermaxfunevals, innermaxfunevals, varargin)

% minimizes FUN in DIM dimensions by multistarts of bbobMinG3PCX.
% maxfunevals is external termination conditions for the outer
% algorithm, not for the inner restarted algorithm

    popsize = 5*DIM + 90;
    fevals = 0;

    % multistart such that ftarget is reached with reasonable prob.
    for ilaunch = 1:100  % relaunch optimizer up to 100 times

        opts.MaxFunEvals = min( outermaxfunevals-fevals, innermaxfunevals );

        % Initialize
        if mod(ilaunch-1, floor(1 + 3 * rand)) == 0
            par = -5 + 10*rand(DIM, popsize); % Random start solution
        else
            par = -0.1 + 0.2*rand(DIM, popsize);
            par = par + repmat(x, 1, popsize); % Start in the neighborhood of the last optimum
        end

        % try minG3mPCX
        [x fmin myfevals exitflag output] = bbobMinG3PCX(FUN, par, ftarget, opts, varargin{:});
        fevals = fevals + myfevals;

        if ...
                (fmin < ftarget) || ...
                (fevals >= outermaxfunevals),
            break;
        end
    end
end
```

Figure 2: Restarting procedure for the G3+PCX algorithm

- after finding a solution with fitness difference $\Delta f \leq 10^{-8}$, or
- after performing more than the allowed number of function evaluations, or
- after the population converged too much, i.e. when all the population members converged to a hypercube with all edges shorter than 10^{-10}.

4. RESULTS

Results from experiments according to [4] on the benchmark functions given in [3, 5] are presented in Figures 3 and 4 and in Table 1. The algorithm found the global optimum (at least once out of 15 runs) for 23, 22, 14, 11, and 11 benchmark functions (out of 24) in 2-, 3-, 5-, 10- and 20-dimensional space, respectively.

5. CPU TIMING EXPERIMENT

The multistart algorithm was run with the maximal number of evaluations set to 10^5, the basic algorithm was restarted for at least 30 seconds. The results show the algorithm takes on approximately 4.1, 4.2, 4.4, 4.7, 5.4, and 7.5 $\times 10^{-4}$ seconds per function evaluation for 2-, 3-, 5-, 10-, 20-, and 40-dimensional search space, respectively. The experiment was conducted on Intel Core 2 CPU, T5600, 1.83 GHz, 1 GB RAM with Windows XP SP3 in MATLAB R2007b.

6. CONCLUSIONS

The generalized generation gap model with the parent centric crossover is an interesting alternative for real-parameter optimization. Its results are best for functions described as *moderate* (including e.g. Rosenbrock function which poses serious trouble for many other types of algorithms). On the other hand, the algorithm completely fails in case of *multimodal* functions. Maybe the effficiency could be improved here by not using the local-search intensive variant of PCX operator.

Acknowledgements

The author is supported by the Grant Agency of the Czech Republic with the grant no. 102/08/P094 entitled “Machine learning methods for solution construction in evolutionary algorithms”.

7. REFERENCES

[1] K. Deb. A population-based algorithm-generator for real-parameter optimization. *Soft Computing*, 9(4):236–253, April 2005.

[2] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. Technical report, Indian Institute of Technology, April 2002.

[3] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking

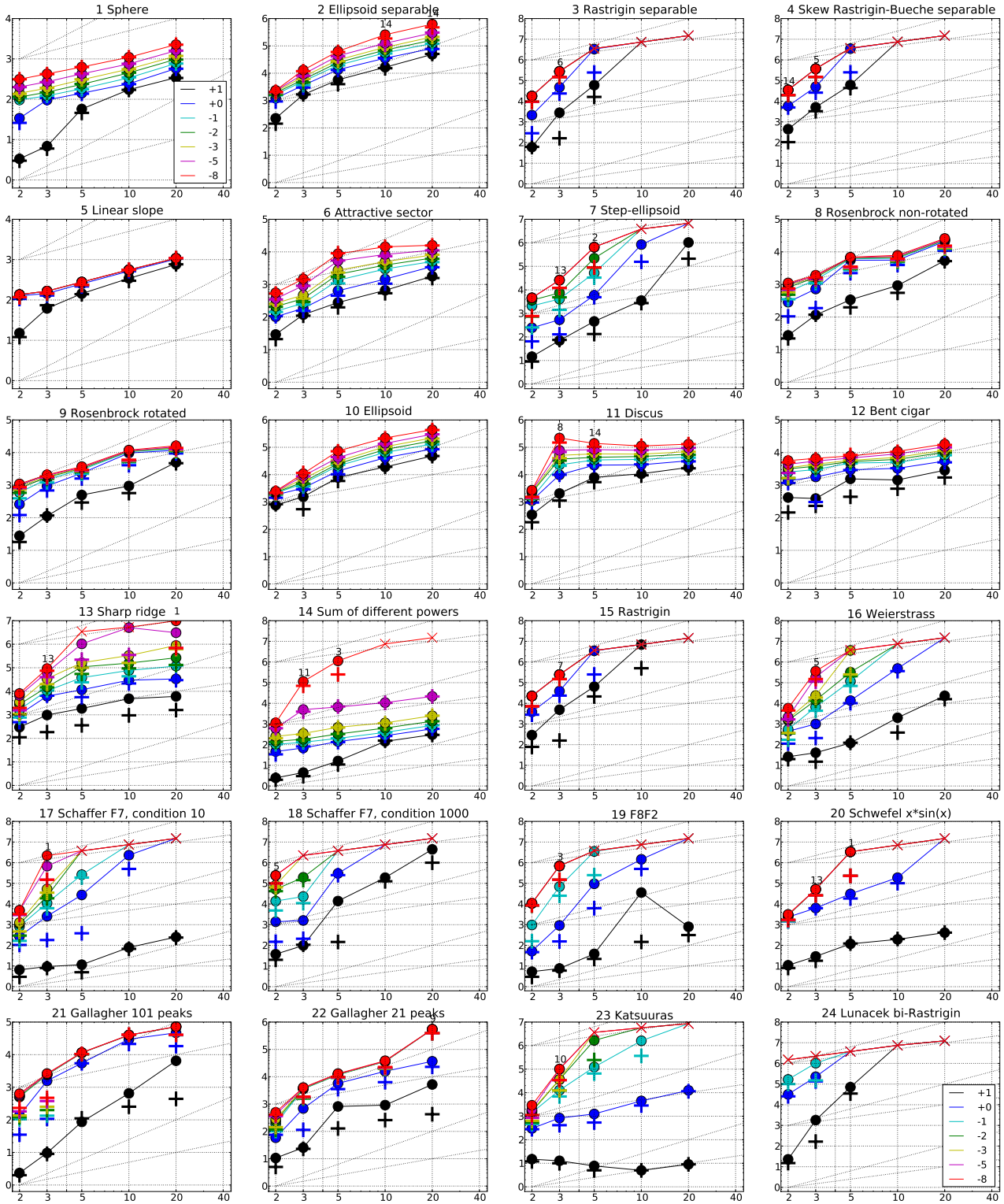

Figure 3: Expected Running Time (ERT, ●) to reach $f_{\text{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. The $\text{ERT}(\Delta f)$ equals to $\#\text{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\text{opt}} + \Delta f$ was surpassed during the trial. The $\#\text{FEs}(\Delta f)$ are the total number of function evaluations while $f_{\text{opt}} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and f_{opt} denotes the optimal function value. Crosses (×) indicate the total number of function evaluations $\#\text{FEs}(-\infty)$. Numbers above ERT-symbols indicate the number of successful trials. Annotated numbers on the ordinate are decimal logarithms. Additional grid lines show linear and quadratic scaling.

f_1 in 5-D, N=15, mFE=757						f_1 in 20-D, N=15, mFE=2402					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	5.7e1	4.5e1	7.1e1	5.7e1	15	3.4e2	3.3e2	3.5e2	3.4e2
1	15	1.4e2	1.4e2	1.5e2	1.4e2	15	5.8e2	5.6e2	5.9e2	5.8e2
1e−1	15	1.8e2	1.7e2	1.9e2	1.8e2	15	7.7e2	7.5e2	8.0e2	7.7e2
1e−3	15	3.0e2	2.9e2	3.2e2	3.0e2	15	1.2e3	1.2e3	1.2e3	1.2e3
1e−5	15	4.3e2	4.1e2	4.4e2	4.3e2	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−8	15	6.3e2	6.1e2	6.5e2	6.3e2	15	2.3e3	2.2e3	2.3e3	2.3e3

f_2 in 5-D, N=15, mFE=130569						f_2 in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	5.7e3	4.4e3	7.0e3	5.7e3	15	4.9e4	4.5e4	5.3e4	4.9e4
1	15	1.3e4	1.1e4	1.6e4	1.3e4	15	8.1e4	7.5e4	8.7e4	8.1e4
1e−1	15	2.0e4	1.7e4	2.2e4	2.0e4	15	1.2e5	1.1e5	1.3e5	1.2e5
1e−3	15	3.1e4	2.6e4	3.5e4	3.1e4	15	2.1e5	2.0e5	2.3e5	2.1e5
1e−5	15	4.3e4	3.7e4	4.9e4	4.3e4	15	3.0e5	2.8e5	3.2e5	3.0e5
1e−8	15	6.4e4	5.6e4	7.2e4	6.4e4	14	6.3e5	5.4e5	7.1e5	6.0e5

f_3 in 5-D, N=15, mFE=250001						f_3 in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	13	6.0e4	3.1e4	9.5e4	5.7e4	0	*13e+1*	*71e+0*	*18e+1*	7.1e4
1	1	3.3e6	3.0e6	3.6e6	2.5e5	.	.	.	.	.
1e−1	0	*30e−1*	*20e−1*	*13e+0*	6.3e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_4 in 5-D, N=15, mFE=250030						f_4 in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	14	6.1e4	4.1e4	8.2e4	5.5e4	0	*19e+1*	*12e+1*	*35e+1*	2.8e5
1	1	3.5e6	3.3e6	3.7e6	2.3e5	.	.	.	.	.
1e−1	0	*50e−1*	*20e−1*	*90e−1*	1.0e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_5 in 5-D, N=15, mFE=577						f_5 in 20-D, N=15, mFE=1586					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.4e2	1.3e2	1.6e2	1.4e2	15	7.6e2	7.1e2	8.1e2	7.6e2
1	15	2.5e2	2.2e2	2.8e2	2.5e2	15	1.0e3	9.4e2	1.1e3	1.0e3
1e−1	15	2.7e2	2.4e2	3.1e2	2.7e2	15	1.1e3	9.8e2	1.2e3	1.1e3
1e−3	15	2.8e2	2.4e2	3.1e2	2.8e2	15	1.1e3	9.9e2	1.2e3	1.1e3
1e−5	15	2.8e2	2.5e2	3.2e2	2.8e2	15	1.1e3	9.9e2	1.2e3	1.1e3
1e−8	15	2.8e2	2.5e2	3.2e2	2.8e2	15	1.1e3	9.8e2	1.2e3	1.1e3

f_6 in 5-D, N=15, mFE=14647						f_6 in 20-D, N=15, mFE=19044					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.8e2	2.1e2	3.5e2	2.8e2	15	1.8e3	1.6e3	2.0e3	1.8e3
1	15	6.4e2	4.6e2	8.5e2	6.4e2	15	3.5e3	3.2e3	3.9e3	3.5e3
1e−1	15	1.3e3	1.0e3	1.6e3	1.3e3	15	5.1e3	4.6e3	5.5e3	5.1e3
1e−3	15	2.7e3	2.2e3	3.2e3	2.7e3	15	8.0e3	7.5e3	8.5e3	8.0e3
1e−5	15	5.3e3	4.5e3	6.1e3	5.3e3	15	1.1e4	1.1e4	1.2e4	1.1e4
1e−8	15	8.7e3	7.4e3	9.9e3	8.7e3	15	1.6e4	1.5e4	1.7e4	1.6e4

f_7 in 5-D, N=15, mFE=110480						f_7 in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	4.6e2	1.5e2	7.7e2	4.6e2	6	1.0e6	7.5e5	1.3e6	2.9e5
1	15	5.9e3	4.1e3	7.9e3	5.9e3	0	*12e+0*	*78e−1*	*22e+0*	1.0e5
1e−1	12	5.2e4	4.0e4	6.4e4	3.8e4	.	.	.	.	.
1e−3	2	6.5e5	6.1e5	6.9e5	8.3e4	.	.	.	.	.
1e−5	2	6.5e5	6.1e5	6.9e5	8.3e4	.	.	.	.	.
1e−8	2	6.5e5	6.1e5	6.9e5	8.3e4	.	.	.	.	.

f_8 in 5-D, N=15, mFE=28431						f_8 in 20-D, N=15, mFE=59710					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.4e2	2.3e2	4.4e2	3.4e2	15	5.4e3	4.9e3	5.8e3	5.4e3
1	15	5.5e3	3.2e3	7.9e3	5.5e3	15	2.1e4	1.6e4	2.6e4	2.1e4
1e−1	15	6.1e3	3.8e3	8.6e3	6.1e3	15	2.2e4	1.7e4	2.8e4	2.2e4
1e−3	15	6.5e3	4.1e3	9.0e3	6.5e3	15	2.3e4	1.8e4	2.8e4	2.3e4
1e−5	15	6.6e3	4.4e3	9.3e3	6.6e3	15	2.4e4	1.9e4	3.0e4	2.4e4
1e−8	15	6.9e3	4.5e3	9.6e3	6.9e3	15	2.6e4	2.1e4	3.1e4	2.6e4

f_9 in 5-D, N=15, mFE=9011						f_9 in 20-D, N=15, mFE=51796					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	5.0e2	3.3e2	6.9e2	5.0e2	15	5.0e3	4.5e3	5.6e3	5.0e3
1	15	2.3e3	1.7e3	2.9e3	2.3e3	15	1.2e4	8.9e3	1.5e4	1.2e4
1e−1	15	3.0e3	2.4e3	3.6e3	3.0e3	15	1.3e4	1.0e4	1.6e4	1.3e4
1e−3	15	3.4e3	2.8e3	3.9e3	3.4e3	15	1.4e4	1.1e4	1.7e4	1.4e4
1e−5	15	3.5e3	2.9e3	4.1e3	3.5e3	15	1.5e4	1.2e4	1.8e4	1.5e4
1e−8	15	3.7e3	3.1e3	4.3e3	3.7e3	15	1.6e4	1.3e4	1.9e4	1.6e4

f_{10} in 5-D, N=15, mFE=148986						f_{10} in 20-D, N=15, mFE=600230					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	7.8e3	5.9e3	9.6e3	7.8e3	15	4.8e4	4.4e4	5.2e4	4.8e4
1	15	1.4e4	1.1e4	1.6e4	1.4e4	15	8.7e4	8.1e4	9.2e4	8.7e4
1e−1	15	2.1e4	1.8e4	2.4e4	2.1e4	15	1.3e5	1.2e5	1.4e5	1.3e5
1e−3	15	3.1e4	2.7e4	3.4e4	3.1e4	15	2.2e5	2.1e5	2.3e5	2.2e5
1e−5	15	4.2e4	3.7e4	4.7e4	4.2e4	15	3.0e5	2.8e5	3.2e5	3.0e5
1e−8	15	7.3e4	6.5e4	8.2e4	7.3e4	15	4.4e5	4.2e5	4.6e5	4.4e5

f_{11} in 5-D, N=15, mFE=250001						f_{11} in 20-D, N=15, mFE=149130					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	7.9e3	5.2e3	1.1e4	7.9e3	15	1.8e4	1.7e4	1.9e4	1.8e4
1	15	2.2e4	1.9e4	2.5e4	2.2e4	15	3.2e4	3.0e4	3.4e4	3.2e4
1e−1	15	3.4e4	3.1e4	3.9e4	3.4e4	15	4.4e4	4.2e4	4.6e4	4.4e4
1e−3	15	5.8e4	5.4e4	6.1e4	5.8e4	15	6.9e4	6.7e4	7.2e4	6.9e4
1e−5	15	8.2e4	7.5e4	9.0e4	8.2e4	15	9.4e4	9.1e4	9.6e4	9.4e4
1e−8	14	1.4e5	1.2e5	1.6e5	1.3e5	15	1.3e5	1.3e5	1.3e5	1.3e5

f_{12} in 5-D, N=15, mFE=14715						f_{12} in 20-D, N=15, mFE=25484					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.6e3	1.1e3	2.0e3	1.6e3	15	2.9e3	2.1e3	3.6e3	2.9e3
1	15	3.0e3	2.3e3	3.6e3	3.0e3	15	5.5e3	4.1e3	6.8e3	5.5e3
1e−1	15	4.8e3	4.2e3	5.3e3	4.8e3	15	8.3e3	6.6e3	9.9e3	8.3e3
1e−3	15	5.7e3	5.0e3	6.5e3	5.7e3	15	1.2e4	1.0e4	1.3e4	1.2e4
1e−5	15	6.8e3	6.0e3	7.8e3	6.8e3	15	1.5e4	1.4e4	1.6e4	1.5e4
1e−8	15	7.9e3	6.7e3	8.9e3	7.9e3	15	1.8e4	1.7e4	1.9e4	1.8e4

f_{13} in 5-D, N=15, mFE=250001						f_{13} in 20-D, N=15, mFE=702788					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.9e3	1.2e3	2.4e3	1.9e3	15	6.1e3	3.3e3	9.5e3	6.1e3
1	15	1.2e4	8.2e3	1.5e4	1.2e4	15	3.4e4	2.4e4	4.3e4	3.4e4
1e−1	15	3.8e4	2.3e4	5.6e4	3.8e4	15	1.2e5	9.9e4	1.4e5	1.2e5
1e−3	11	1.6e5	1.2e5	2.0e5	1.1e5	9	8.8e5	7.7e5	9.9e5	4.8e5
1e−5	3	1.0e6	9.5e5	1.1e6	2.1e5	3	3.1e6	2.8e6	3.4e6	5.3e5
1e−8	0	*15e−5*	*20e−7*	*11e−3*	1.0e5	1	9.9e6	9.3e6	1.0e7	6.8e5

f_{14} in 5-D, N=15, mFE=250001						f_{14} in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.6e1	1.0e1	2.3e1	1.6e1	15	3.0e2	2.9e2	3.2e2	3.0e2
1	15	1.4e2	1.3e2	1.5e2	1.4e2	15	5.8e2	5.6e2	6.0e2	5.8e2
1e−1	15	2.0e2	1.9e2	2.1e2	2.0e2	15	8.5e2	8.2e2	8.9e2	8.5e2
1e−3	15	6.9e2	6.5e2	7.4e2	6.9e2	15	2.5e3	2.4e3	2.6e3	2.5e3
1e−5	15	6.6e3	5.7e3	7.6e3	6.6e3	15	2.2e4	2.0e4	2.3e4	2.2e4
1e−8	3	1.1e6	1.0e6	1.2e6	2.5e5	0	*82e−9*	*71e−9*	*12e−8*	7.9e5

f_{15} in 5-D, N=15, mFE=250001						f_{15} in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	13	6.4e4	3.4e4	9.5e4	5.7e4	0	*92e+0*	*59e+0*	*13e+1*	2.0e5
1	1	3.5e6	3.3e6	3.7e6	2.5e5	.	.	.	.	.
1e−1	0	*50e−1*	*20e−1*	*11e+0*	7.1e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_{16} in 5-D, N=15, mFE=250010						f_{16} in 20-D, N=15, mFE=1000160					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.2e2	1.0e2	1.4e2	1.2e2	15	2.3e4	1.4e4	3.2e4	2.3e4
1	15	1.4e4	1.0e4	1.7e4	1.4e4	0	*32e−1*	*20e−1*	*48e−1*	5.0e5
1e−1	13	1.2e5	8.6e4	1.5e5	9.9e4	.	.	.	.	.
1e−3	1	3.6e6	3.5e6	3.8e6	2.5e5	.	.	.	.	.
1e−5	0	*60e−4*	*19e−4*	*12e−2*	1.3e5	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_{17} in 5-D, N=15, mFE=250001						f_{17} in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.2e1	8.1e0	1.6e1	1.2e1	15	2.5e2	2.2e2	2.8e2	2.5e2
1	14	2.8e4	8.1e3	5.0e4	2.6e4	0	*35e−1*	*22e−1*	*49e−1*	3.2e5
1e−1	9	2.6e5	2.1e5	3.2e5	1.5e5	.	.	.	.	.
1e−3	0	*60e−3*	*44e−3*	*36e−2*	1.0e5	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_{18} in 5-D, N=15, mFE=250001						f_{18} in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.4e4	5.2e3	2.5e4	1.4e4	3	4.4e6	3.8e6	5.0e6	1.0e6
1	7	3.0e5	2.1e5	3.9e5	1.8e5	0	*14e+0*	*71e−1*	*20e+0*	4.5e5
1e−1	0	*12e−1*	*17e−2*	*47e−1*	5.6e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_{19} in 5-D, N=15, mFE=250001						f_{19} in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.9e1	2.8e1	5.0e1	3.9e1	15	8.0e2	5.1e2	1.1e3	8.0e2
1	11	9.5e4	4.9e4	1.4e5	4.9e4	0	*26e−1*	*15e−1*	*46e−1*	8.9e5
1e−1	1	3.5e6	3.3e6	3.8e6	2.5e5	.	.	.	.	.
1e−3	0	*50e−2*	*27e−2*	*15e−1*	1.4e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_{20} in 5-D, N=15, mFE=250000						f_{20} in 20-D, N=15, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.2e2	1.1e2	1.3e2	1.2e2	15	4.1e2	4.0e2	4.3e2	4.1e2
1	15	3.1e4	2.1e4	4.2e4	3.1e4	0	*12e−1*	*11e−1*	*13e−1*	4.5e5
1e−1	1	3.4e6	3.3e6	3.5e6	2.1e5	.	.	.	.	.
1e−3	1	3.4e6	3.3e6	3.5e6	2.1e5	.	.	.	.	.
1e−5	1	3.4e6	3.3e6	3.5e6	2.1e5	.	.	.	.	.
1e−8	1	3.4e6	3.3e6	3.5e6	2.1e5	.	.	.	.	.

f_{21} in 5-D, N=15, mFE=27363						f_{21} in 20-D, N=15, mFE=309896					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	8.6e1	7.1e1	1.0e2	8.6e1	15	6.5e3	2.4e3	1.1e4	6.5e3
1	15	5.5e3	4.1e3	7.0e3	5.5e3	15	4.7e4	2.7e4	6.6e4	4.7e4
1e−1	15	1.1e4	8.4e3	1.4e4	1.1e4	15	7.1e4	4.2e4	9.8e4	7.1e4
1e−3	15	1.1e4	8.9e3	1.4e4	1.1e4	15	7.2e4	4.4e4	1.0e5	7.2e4
1e−5	15	1.1e4	8.9e3	1.4e4	1.1e4	15	7.2e4	4.4e4	1.0e5	7.2e4
1e−8	15	1.2e4	8.9e3	1.4e4	1.2e4	15	7.3e4	4.5e4	1.0e5	7.3e4

f_{22} in 5-D, N=15, mFE=31894						f_{22} in 20-D, N=15, mFE=605212					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	8.2e2	1.3e2	1.5e3	8.2e2	15	5.2e3	1.7e3	9.1e3	5.2e3
1	15	5.7e3	3.1e3	8.5e3	5.7e3	15	3.7e4	2.5e4	4.9e4	3.7e4
1e−1	15	1.2e4	8.6e3	1.5e4	1.2e4	9	5.4e5	4.3e5	6.5e5	3.2e5
1e−3	15	1.2e4	9.1e3	1.6e4	1.2e4	9	5.4e5	4.4e5	6.5e5	3.2e5
1e−5	15	1.3e4	9.5e3	1.6e4	1.3e4	9	5.4e5	4.3e5	6.5e5	3.3e5
1e−8	15	1.3e4	9.8e3	1.6e4	1.3e4	9	5.5e5	4.4e5	6.6e5	3.3e5

f_{23} in 5-D, N=15, mFE=250001						f_{23} in 20-D, N=15, mFE=602624					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	7.7e0	5.1e0	1.1e1	7.7e0	15	9.0e0	7.1e0	1.1e1	9.0e0
1	15	1.2e3	7.8e2	1.7e3	1.2e3	15	1.3e4	9.7e3	1.6e4	1.3e4
1e−1	12	1.2e5	8.9e4	1.6e5	1.0e5	0	*30e−2*	*20e−2*	*36e−2*	3.2e5
1e−3	0	*44e−3*	*78e−4*	*15e−2*	1.1e5	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

f_{24} in 5-D, N=15, mFE=250009						f_{24} in 20-D, N=13, mFE=1000000					
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	13	7.1e4	4.3e4	1.0e5	6.1e4	0	*97e+0*	*71e+0*	*15e+1*	2.8e5
1	0	*61e−1*	*35e−1*	*10e+0*	6.3e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: Shown are, for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\text{opt}}+\Delta f$ (ERT, see Figure 3); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value (RT_{succ}). If $f_{\text{opt}}+\Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 3 for the names of functions.

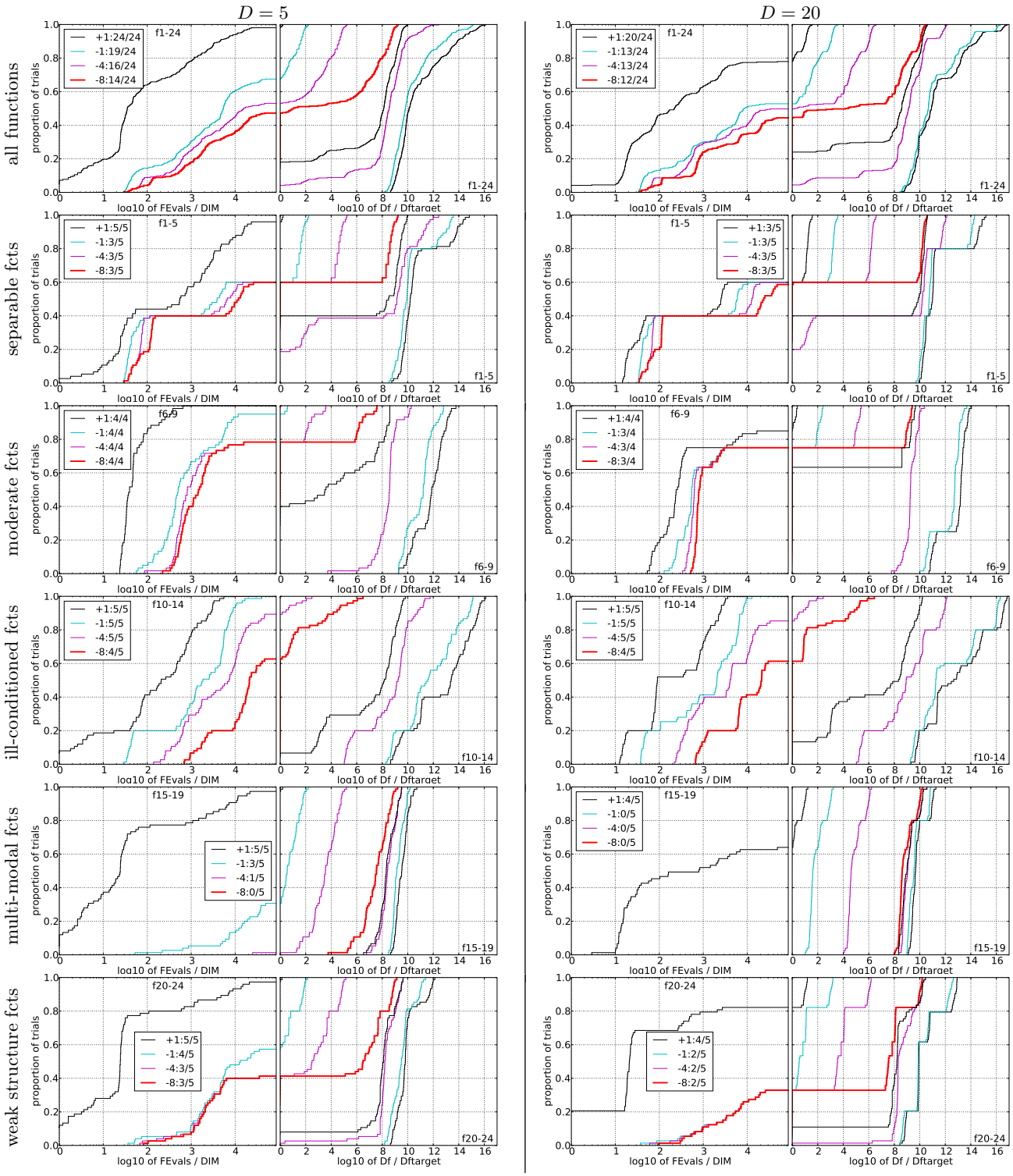

Figure 4: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the best achieved Δf divided by 10^k (upper left lines in continuation of the left subplot), and best achieved Δf divided by 10^{-8} for running times of $D, 10\,D, 100\,D \ldots$ function evaluations (from right to left cycling black-cyan-magenta). Top row: all results from all functions; second row: separable functions; third row: misc. moderate functions; fourth row: ill-conditioned functions; fifth row: multi-modal functions with adequate structure; last row: multi-modal functions with weak structure. The legends indicate the number of functions that were solved in at least one trial. FEvals denotes number of function evaluations, D and DIM denote search space dimension, and Δf and Df denote the difference to the optimal function value.

2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009.

[4] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.

[5] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009.

[6] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.